프로젝트 기획서

(포트폴리오 용도)

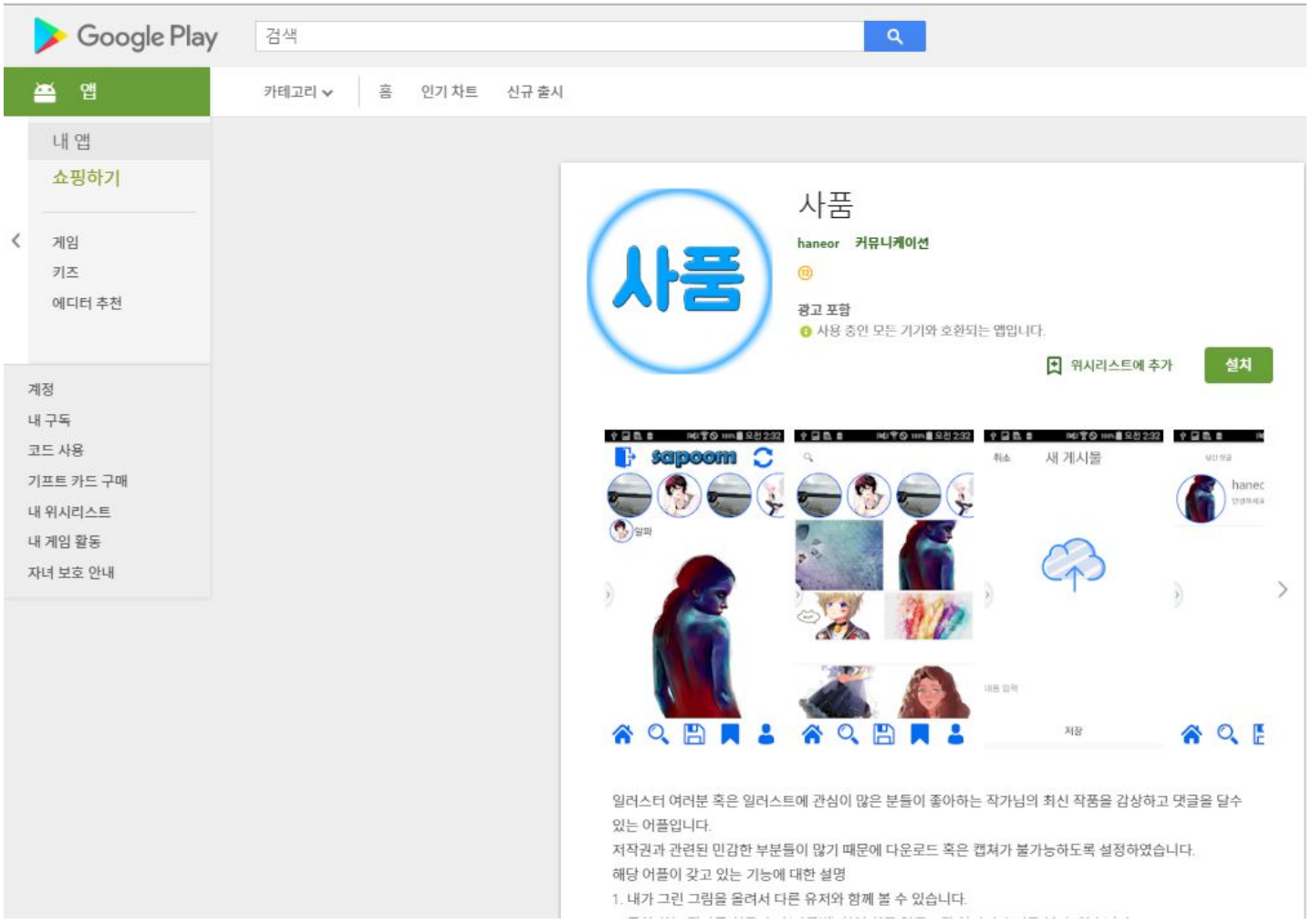
훈련 과정	자바(JAVA)를 활용한 안드로이드 앱(APP)
개발 기간	2018.7.1 ~ 8.10
APP 명칭	사 품

작성자 : 박 한 얼

App 설명 (유사 인스타그램)

일러스트레이터 및 일러스트레이터를 지망하는 학생등... 많은 사람들이 일러스트를 보며, 서로의 작품에 댓글을 달고, 친구(따름벗)를 설정하고 최신 업데이트 내용을 메인 화면에서 확인 할 수 있습니다.

- 제작한 App을 구글 스토어에 등록하였습니다.



화면 구성

순서	화면명
1	스플래쉬 및 로그인 화면 = LoginActivity
2	메인 (팔로우 업데이트) 화면 = MainActivity
3	검색 (최신 업데이트) 화면 = SearchActivity
4	이미지 업로드(Put Edit) 화면 = PuteditActivity
5	북마크 (댓글 확인) 화면 = Bookmark Activity
6	개인 정보 변경 및 확인 화면 = Mine Activity

화면 구조도

로그인 화면

로그인 화면 설명

1. App을 실행하였을 때 1초의 딜레이가 발생
2. 회원가입 데이터 입력 후 아이디의 중복을 작성된 php로 판정 후 회원가입 데이터를 DB에 저장
3. 로그인 데이터를 입력받아 작성된 php를 통해 데이터를 판정 후 액티비티 전환

핵심 코드

- 영문 및 숫자만 입력 가능하도록 하는 메소드

```java
// 영문 및 숫자만 입력이 가능하도록 하는 메소드 생성
public InputFilter filterAlphaNumber = (source, start, end, dest, dstart, dend) -> {
        Pattern ps = Pattern.compile("^[a-zA-Z0-9]+$");
        if (!ps.matcher(source).matches()) {
            return "";
        }
        return null;
};
```

- 회원가입 관련 메소드 joinID

```java
public void joinID () {
    if(!validate) {
        AlertDialog.Builder builder = new AlertDialog.Builder( context: SignupActivity.this);
        builder.setMessage("먼저 중복 체크를 해주세요.").setNegativeButton( text: "확인",  listener: null).show();
        return;
    }

    Response.Listener<String> responseListener = (response) -> {
            try {
                JSONObject jsonResponse = new JSONObject(response);
                boolean success = jsonResponse.getBoolean( name: "success");
                if(success) {
                    AlertDialog.Builder builder = new AlertDialog.Builder( context: SignupActivity.this);
                    builder.setMessage("회원가입에 성공하였습니다.").setPositiveButton( text: "확인",  listener: null).show();
                    finish();
                }else {
                    AlertDialog.Builder builder = new AlertDialog.Builder( context: SignupActivity.this);
                    builder.setMessage("회원가입에 실패하셨습니다.").setPositiveButton( text: "확인",  listener: null).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
    };
    RegisterRequest registerRequest = new RegisterRequest( db_nickname, db_id, db_pw, responseListener);
    RequestQueue queue = Volley.newRequestQueue( context: SignupActivity.this);
    queue.add(registerRequest);
}
}
```

- Splash 화면에서 공지가 없을 시 1초 딜레이 후 화면 전환

```java
    if(splash_message != null) {
        AlertDialog.Builder builder = new AlertDialog.Builder( context: this);
        builder.setMessage(splash_message).setPositiveButton( text: "확인", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                finish();
            }
        });
        builder.create().show();
    }else {
        runnable = (Runnable) () -> {
                startActivity(new Intent( packageContext: SplashActivity.this, LoginActivity.class));
                finish();
        };
        handler = new Handler();
        handler.postDelayed(runnable,  delayMillis: 1000);
    }

}
```

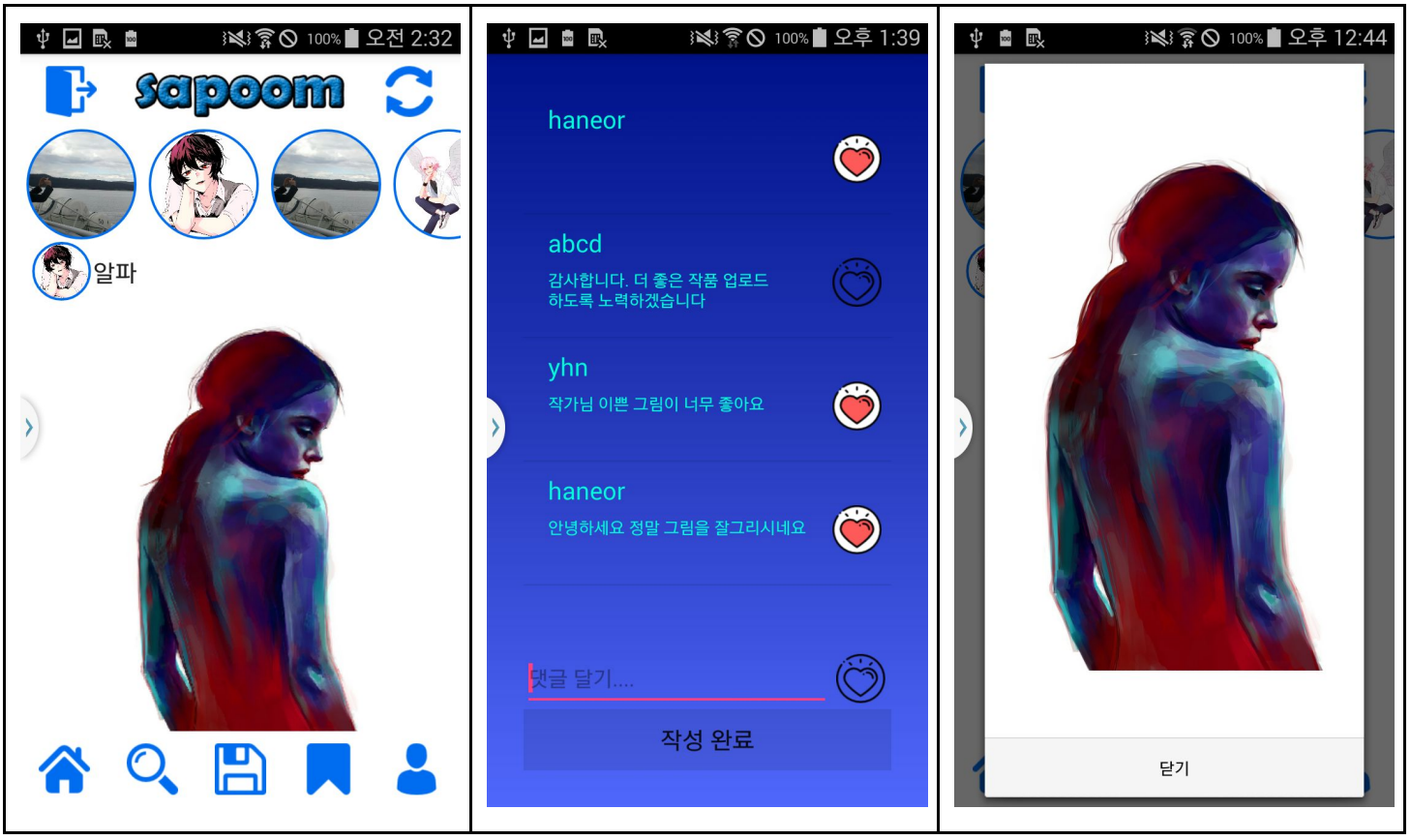
- PHP & MySQL 문서

```php
<?php
header('Content-Type: application/json; charset=utf8');
$conn = mysqli_connect("localhost", "eor0601", "haneor8861", "eor0601");

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

if (!$conn->set_charset("utf8")) {
    exit();
} else {}

$db_id = $_POST["db_id"];
$db_pw = $_POST["db_pw"];
$db_nickname = $_POST["db_nickname"];

$sql = "INSERT INTO users (db_id, db_pw, db_nickname)
VALUES ('$db_id', '$db_pw', '$db_nickname');";

if (mysqli_multi_query($conn, $sql)) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
$response = array();
$response["success"] ="true";

echo json_encode($response);

mysqli_close($conn);
?>
```

조금 더 자세한 내용은 마지막에 기재하였습니다.

메인 화면

(팔로우 한 유저의 최신 업데이트 정보를 확인)



메인 화면 설명

1. 가로 ListView는 최근 업데이트 한 모든 유저의 프로필 사진을 라운드 처리하여 출력
2. 프로필 사진을 원클릭 시 친구 추가 다이얼로그 출력
3. php 를 이용한 DB데이터 통신 및 유저의 사진 정보를 화면에 출력하고 댓글/좋아요 등 변경되는 정보들을 전송
4. 세로 ListView의 경우 친구(따름벗) 설정된 유저의 최신 정보를 출력

핵심 코드

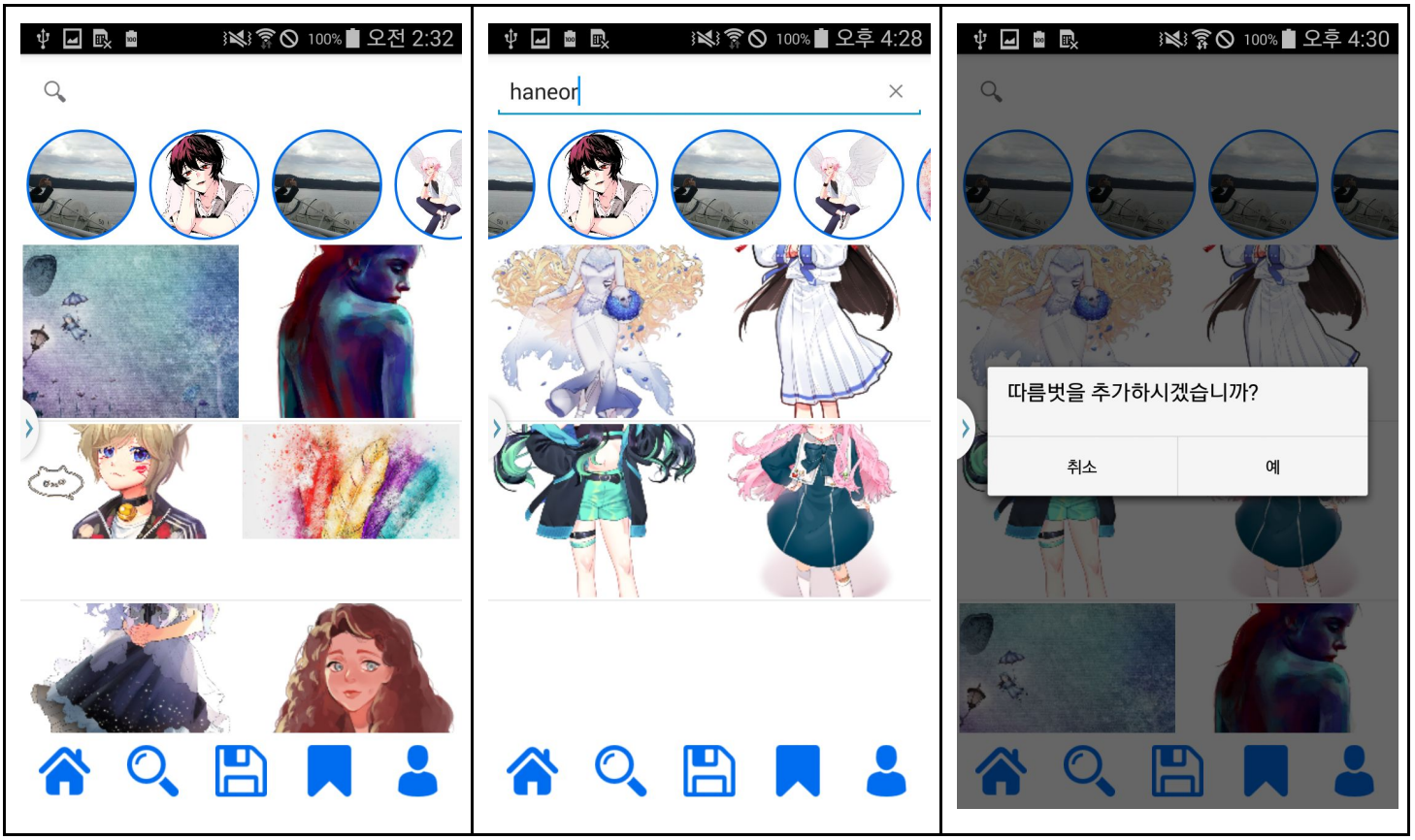
- ListView의 이미지를 클릭하면 다이얼로그로 이미지가 팝업

```java
public void showDialog(String url) {
    android.app.AlertDialog.Builder builder;
    android.app.AlertDialog alertDialog;

    LayoutInflater inflater = (LayoutInflater) this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    View layout = inflater.inflate(R.layout.dialog_user_image, (ViewGroup) findViewById(R.id.layout_root));
    ImageView image = (ImageView) layout.findViewById(R.id.imageView);
    Picasso.with(this).load(url).into(image);
    builder = new android.app.AlertDialog.Builder( context: this);
    builder.setView(layout);
    builder.setNegativeButton( text: "닫기", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
        }
    });
    alertDialog = builder.create();
    alertDialog.show();
}
```

- 가로 ListView 처리 시 이미지 용량 축소 및 이미지 호출

```java
@Override
public void onBindViewHolder(ViewHolder viewHolder, final int i) {
    final Thread mThread = run() -> {
            try {
                URL url = new URL(profile_image_horizelist.get(i));
                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setDoInput(true);
                conn.connect();
                InputStream is = conn.getInputStream();
                bitmap = BitmapFactory.decodeStream(is);
                final Handler handler = new Handler();
                handler.postDelayed(() -> {
                        //bitmap = BitmapFactory.decodeStream(is);
                }, delayMillis: 100);
                int height = bitmap.getHeight();
                int width = bitmap.getWidth();
                if(width>1000){
                    bitmap = Bitmap.createScaledBitmap(bitmap, (width=100), (height = height*100/width), filter: true);
                    bitmap.compress(Bitmap.CompressFormat.PNG, quality: 70, out); }
            }catch (Exception e){ e.printStackTrace(); }
    };
    mThread.start();
    try {
        mThread.join();
        viewHolder.imgThumbnail.setImageBitmap(bitmap);
    } catch (Exception e) {e.printStackTrace();}
```

검색 화면



검색 화면 설명

1. 최신 업데이트되는 모든 유저의 일러스트를 php를 이용해 DB에서 가져와 호출
2. searchView를 이용해서 유저의 아이디를 검색 할 수 있고, 해당 이미지를 ListView에 호출
3. 메인 화면과 연동된 가로 ListView의 프로필 사진을 원크릭 시 다이얼로그 출력 및 승인 시 DB 서버로 데이터 저장

핵심 코드

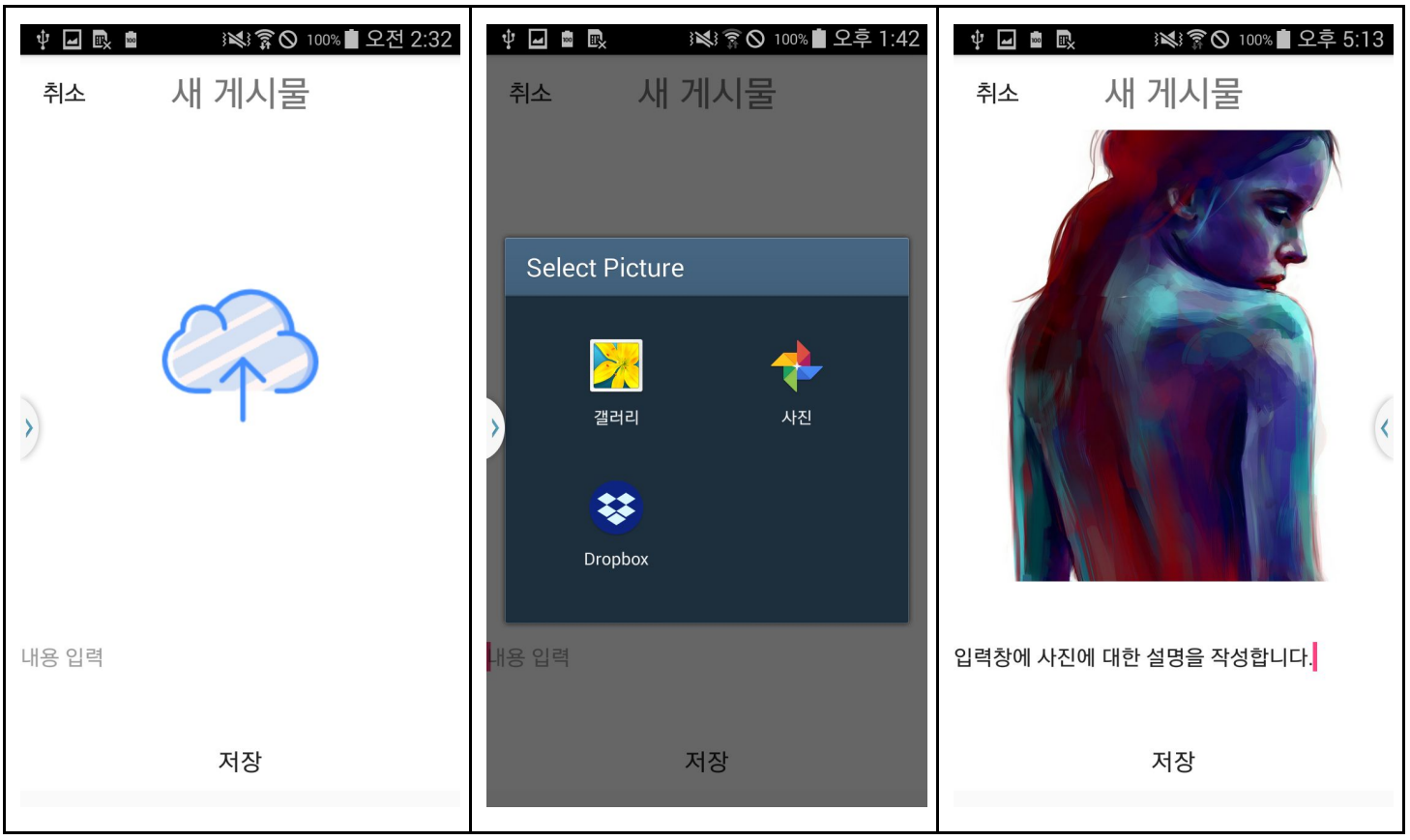
- 검색 결과로 얻어진 사진 주소를 ListView로 호출

```java
private void setSearchView() {
    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String s) {
            hotAdapter = new HotAdapter(getApplicationContext());
            for(int i = 0; i<imageArray.size(); i++) {
                if(s.equals(id.get(i))) {
                    if(i%2 == 0) {
                        evenImage = IMAGE_URL+imageArray.get(i);
                        continue;
                    }else {
                        oddImage = IMAGE_URL+imageArray.get(i);
                        hotAdapter.addItem(new NewestImage_Item(evenImage, oddImage));
                    }
                    newestImageList.setAdapter(hotAdapter);
                }
            }
            return false;
        }

        @Override
        public boolean onQueryTextChange(String s) {
            return false;
        }
    });
}
```

- 서버에 저장된 사진 정보를 최신순서로 가져오는 메소드

```java
private void onLoadUpDateItem () {
    RequestQueue requestQueue = Volley.newRequestQueue( context: this);
    StringRequest stringRequest = new StringRequest(Request.Method.POST, url: "http://eor0601.dothome.co.kr/sel_newestImage.php",
            new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                oddImage = null;
                evenImage = null;
                JSONObject jsonObject_response = new JSONObject(response);
                JSONArray jsonArray = (JSONArray) jsonObject_response.get("response");
                for(int i = 0; i<jsonArray.length(); i++) {
                    id.add(jsonArray.getJSONObject(i).getString( name: "db_id"));
                    imageArray.add(jsonArray.getJSONObject(i).getString( name: "image_id"));
                    if(i%2 == 0) {
                        evenImage = IMAGE_URL+imageArray.get(i);
                        continue;
                    }else {
                        oddImage = IMAGE_URL+imageArray.get(i);
                        hotAdapter.addItem(new NewestImage_Item(evenImage, oddImage));
                    }
                    newestImageList.setAdapter(hotAdapter);
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

        }
```

이미지 업로드 화면



이미지 업로드 화면 설명

1. 이미지뷰를 클릭하면 갤러리에서 이미지를 선택해서 호출
2. 이미지, 이미지 설명, 날짜 정보등을 php로 DB에 저장

핵심 코드

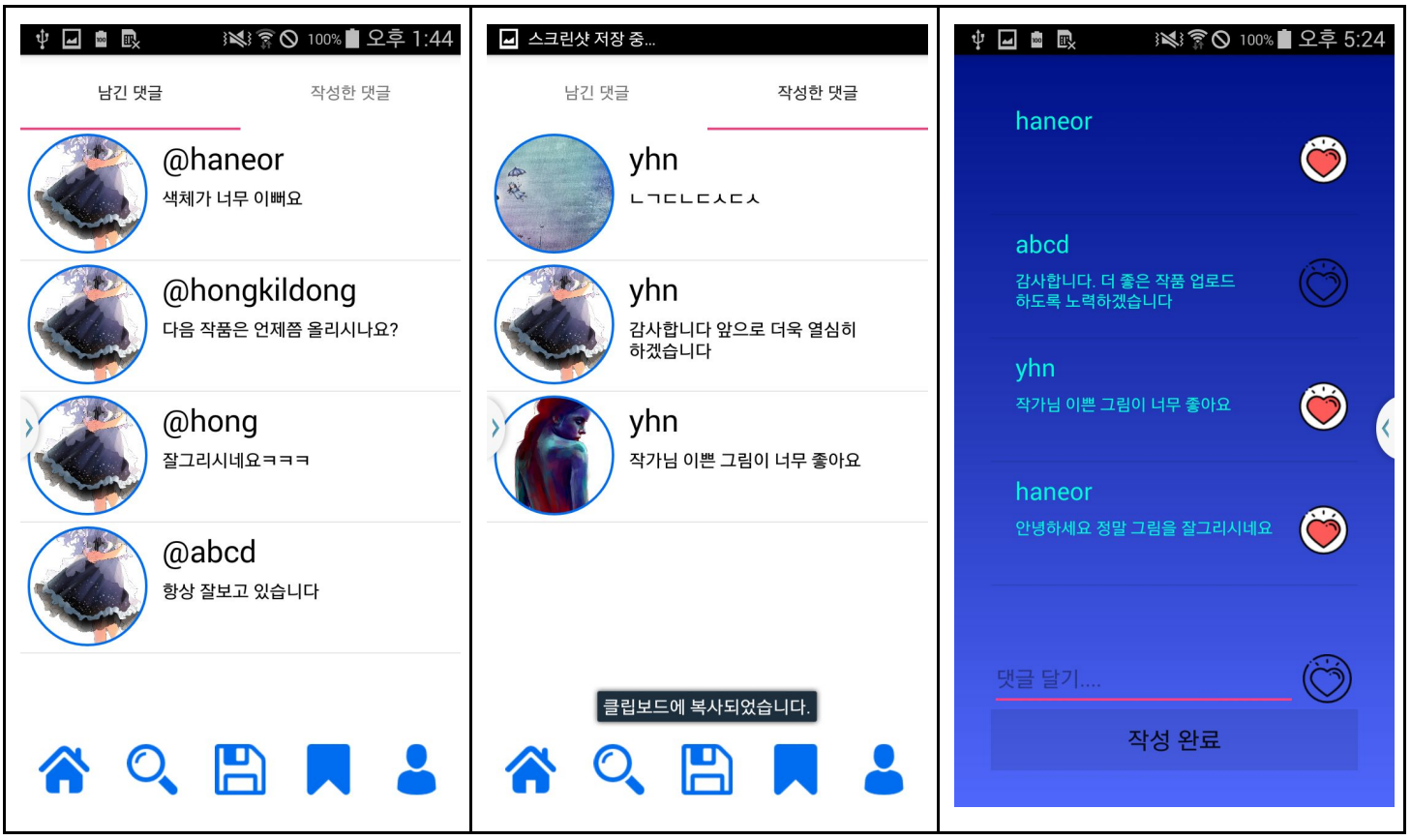
- Gallery 에서 사진 데이터를 화면에 출력

```java
public void loadImagefromGallery(View view) {
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    intent.setType("image/*");
    startActivityForResult(Intent.createChooser(intent, title: "Select Picture"), PICK_IMAGE_REQUEST);
}

//이미지 선택작업을 후의 결과 처리
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    try {
        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && null != data) {
            Uri uri = data.getData();
            Bitmap bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(),uri);
            //이미지가 한계이상 크면 불러 오지 못하므로 사이즈를 줄여 준다.
            int nh = (int) (bitmap.getHeight() * (800.0 / bitmap.getWidth()));
            Bitmap scaled = Bitmap.createScaledBitmap(bitmap, dstWidth: 800, nh, filter: true);

            imageBtn = findViewById(R.id.imageBtn);
            imageBtn.setImageBitmap(scaled);
            uploadBitmap(scaled);
        } else {
            Toast.makeText( context: this, text: "취소 되었습니다.", Toast.LENGTH_LONG).show();
        }
    } catch (Exception e) {
        Toast.makeText( context: this, text: "로딩에 오류가 있습니다.", Toast.LENGTH_LONG).show();
        e.printStackTrace();
    }
}
```

- VolleyMultipartRequest를 이용한 DB서버로 사진 저장

```java
private void uploadBitmap(final Bitmap bitmap) {
    VolleyMultipartRequest volleyMultipartRequest = new VolleyMultipartRequest(Request.Method.POST, BASE_URL,
            new Response.Listener<NetworkResponse>() {
                @Override
                public void onResponse(NetworkResponse response) {
                    try {
                        // 프로필 수정 시 업로드된 사진을 피드백 받아서 설정함.
                        String responsePath = new String(response.data, charsetName: "UTF-8");
                        // 저장 시 버그로 ""의 포함으로 버그 발생하여 처리함.
                        image_id = responsePath.substring(1, 37);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    // 만약 업데이트 사진의 주소를 확보 하려면 IMAGE_URL + image_id (String) 으로 받아 오면 됨.
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Toast.makeText(getApplicationContext(), error.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }) {

        protected Map<String, String> getParams() {...}

        protected Map<String, DataPart> getByteData() {
            Map<String, DataPart> params = new HashMap<>();
            long imagename = System.currentTimeMillis();
            params.put( k: "upfile", new DataPart( name: imagename + ".png", getFileDataFromDrawable(bitmap)));
            return params;
        }
    };
    Volley.newRequestQueue( context: this).add(volleyMultipartRequest);
```

북마크 화면



북마크 화면 설명

1. 메인화면에서 남긴 댓글의 내용을 확인 하기 위한 ListView를 활성 (내가 남긴 댓글 / 내가 작성한 댓글)
2. php를 이용해서 댓글의 이미지와 내용을 저장하고 호출
3. php를 이용해 내가 작성한 댓글을 삭제

핵심 코드

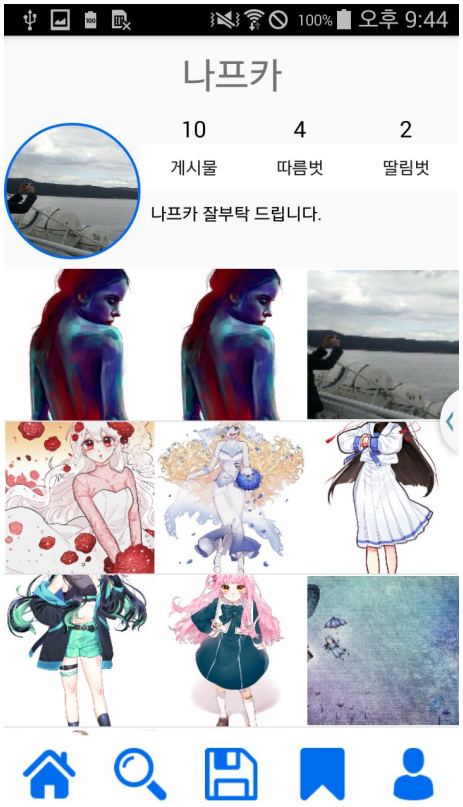
- 프래그 먼트를 이용한 화면 전환 및 탭 버튼 사용

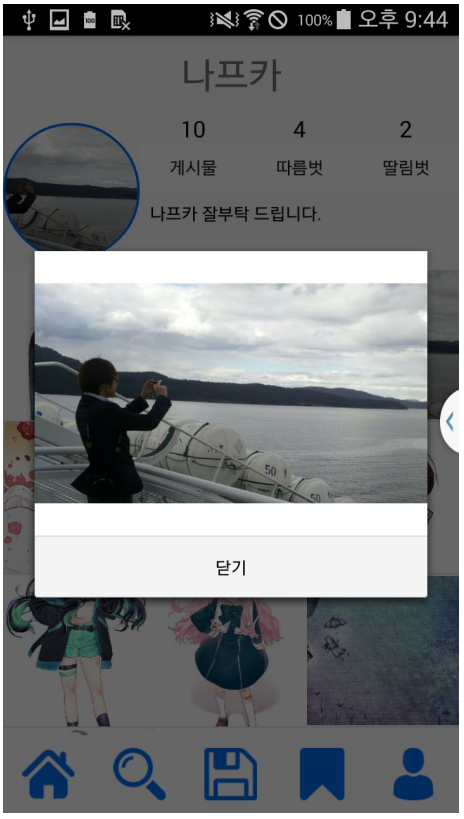
```
final TabLayout tabs = (TabLayout) findViewById(R.id.tabs);
tabs.addTab(tabs.newTab().setText(follow_comment));
tabs.addTab(tabs.newTab().setText(dogging_comment));

tabs.setOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override
    public void onTabSelected(TabLayout.Tab tab) {
        if(tab.getText().equals(follow_comment)) {
            getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment_do).commit();
        }else {
            getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment_book).commit();
        }
    }
    @Override
    public void onTabUnselected(TabLayout.Tab tab) {   }
    @Override
    public void onTabReselected(TabLayout.Tab tab) {   }
});
```

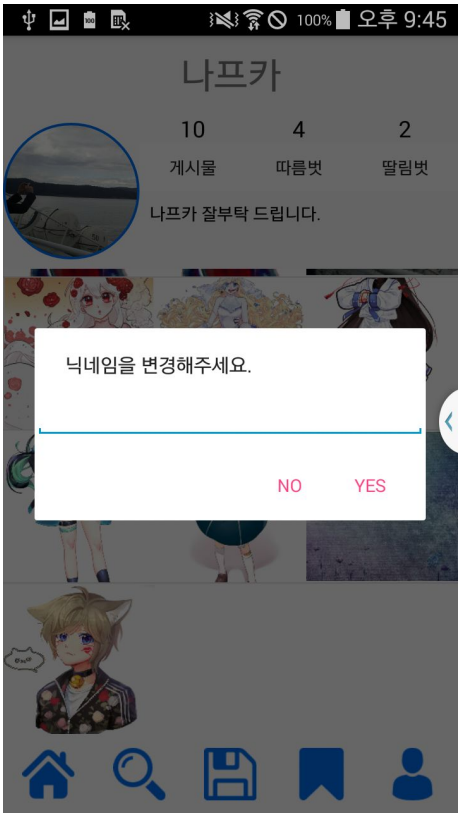
- 입력된 댓글의 내용을 DB 서버로 저장하는 메소드

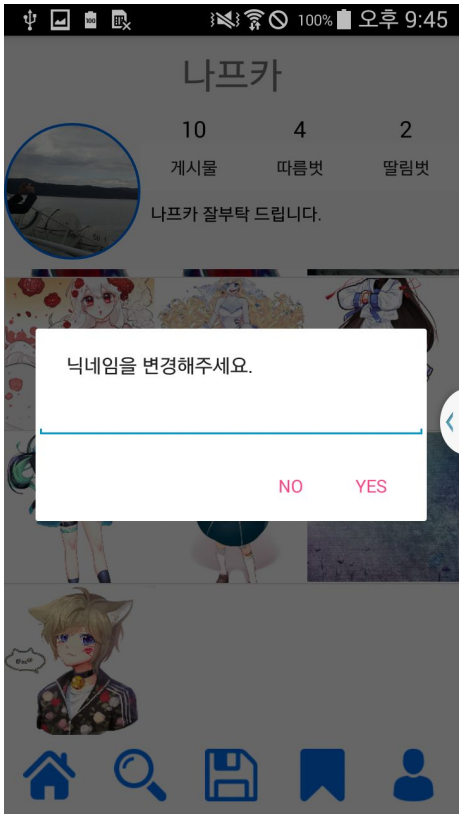
```
private void setEditTextItem () {
    String index = "0";
    String created = "date('Y-m-d H:i:s')";
    user_id = LoginActivity.db_id;
    Response.Listener<String> responseListener = new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {     }
    };
    CommentInsertRequest commentInsertRequest = new CommentInsertRequest(index, image_id, comment_sub,
            user_id, good, created, responseListener);
    RequestQueue queue = Volley.newRequestQueue( context: CommentActivity.this);
    queue.add(commentInsertRequest);
}
```

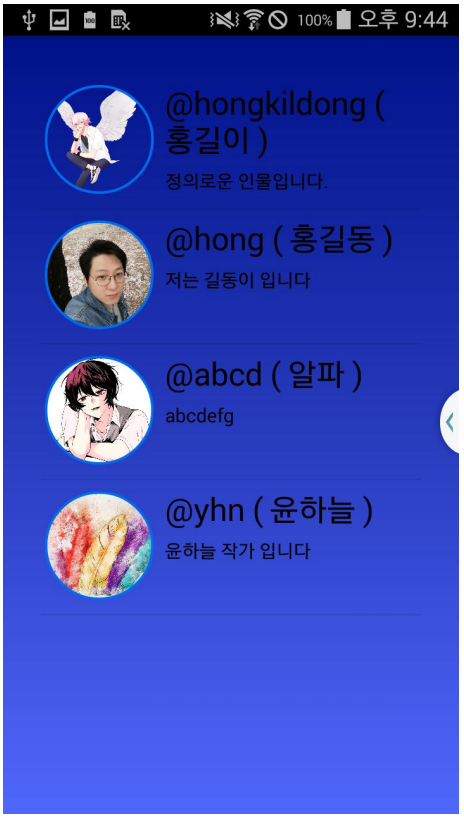
개인 정보 화면

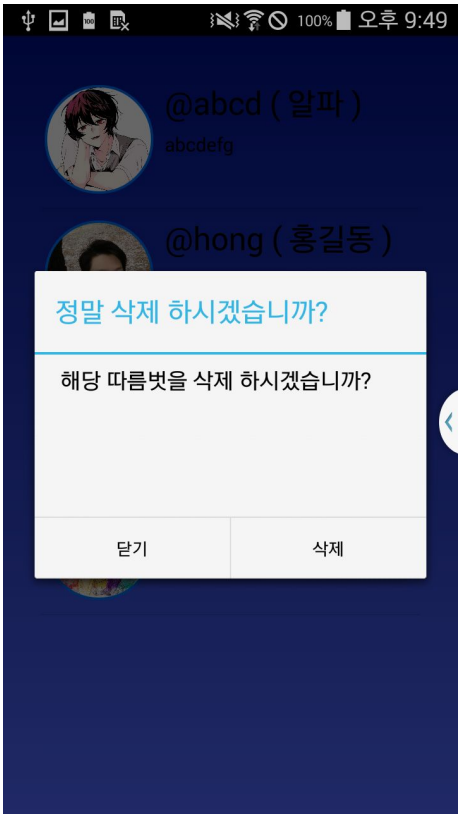

오후 9:44
나프카
10
게시물
4
따름벗
2
딸림벗
나프카 잘부탁 드립니다.


오후 9:44
나프카
10
게시물
4
따름벗
2
딸림벗
나프카 잘부탁 드립니다.
닫기


오후 9:45
나프카
10
게시물
4
따름벗
2
딸림벗
나프카 잘부탁 드립니다.
닉네임을 변경해주세요.
NO
YES


오후 9:45
나프카
10
게시물
4
따름벗
2
딸림벗
나프카 잘부탁 드립니다.
닉네임을 변경해주세요.
NO
YES


오후 9:44
@hongkildong (홍길이)
정의로운 인물입니다.
@hong (홍길동)
저는 길동이 입니다
@abcd (알파)
abcdefg
@yhn (윤하늘)
윤하늘 작가 입니다


오후 9:49
@abcd (알파)
abcdefg
@hong (홍길동)
정말 삭제 하시겠습니까?
해당 따름벗을 삭제 하시겠습니까?
닫기
삭제

개인 정보 화면 설명

1. 개인 프로필 정보 변경 및 변경된 사항을 php를 통해 DB에 저장
2. 프로필 이미지 DB 서버에 저장 및 이미지 용량 변경
3. 업로드한 이미지를 리스트뷰에 호출
4. 호출된 이미지를 클릭 시 다이얼로그를 이용해 확인
5. 친구(따름벗/딸림벗) 버튼 클릭 시 해당 목록을 리스트뷰로 확인 하고 삭제

핵심 코드

- CircleImageView에 이미지의 url 주소를 통해 호출

```
private void downLoadImageURL (final String imageUrl) {
    final Handler handler = new Handler();
    //Thread t = new Thread(Runnable 객체를 만든다);
    Thread t = new Thread(run() -> {    // 오래 거릴 작업을 구현한다
            try {
                final CircleImageView iv = (CircleImageView) findViewById(R.id.profileImageBtn);
                URL url = new URL(imageUrl);
                InputStream is = url.openStream();
                final Bitmap bm = BitmapFactory.decodeStream(is);
                handler.post(run() -> {  // 화면에 그려줄 작업
                        iv.setImageBitmap(bm);
                });
                iv.setImageBitmap(bm); //비트맵 객체로 보여주기
            } catch (Exception e) { }
    });
    t.start();
}
```

- DB로 부터 받은 이미지의 url주소를 ListView 1행당 3개씩 넣어주는 조건문

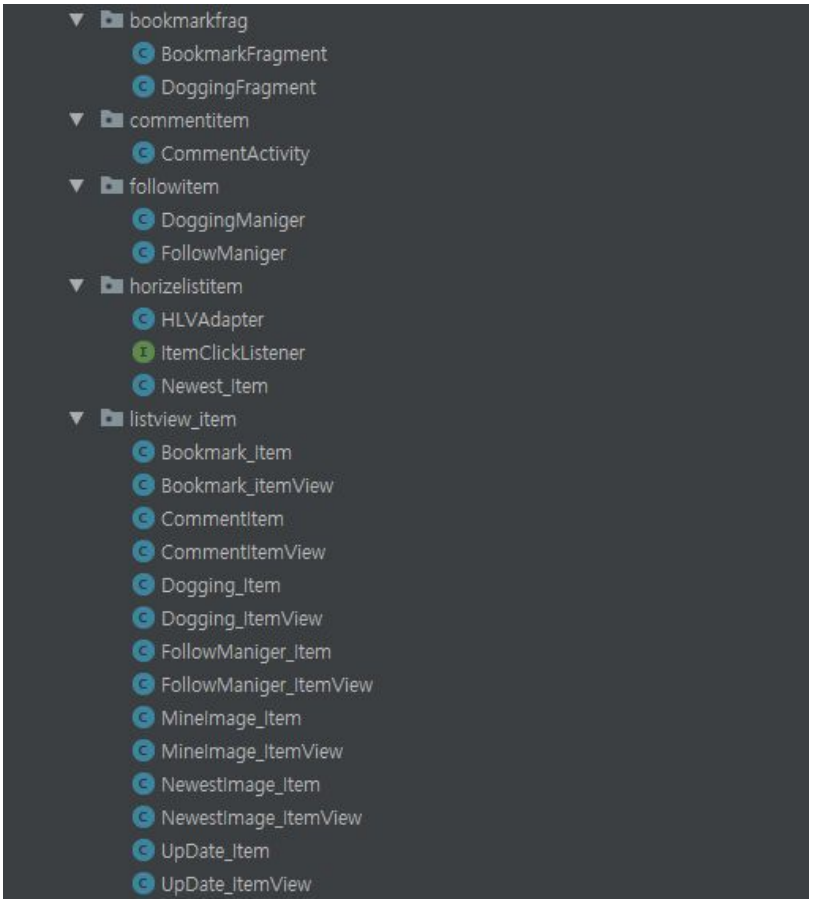
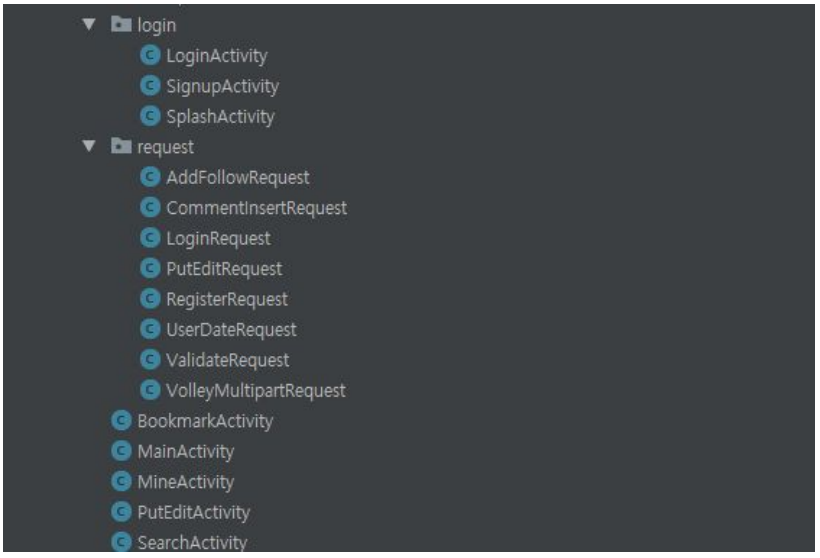
```
private void onListImageSet (int arrayLength) {
    MineUserAdapter mineAdapter = new MineUserAdapter(getApplicationContext());
    String [] divItem = new String[3];
    for (int i = 0; i < arrayLength; i++) {
        if(i%3 == 0) {
            divItem[0] = arrayStr[i];
            if(i==arrayLength-1) {
                divItem[1] = "http://eor0601.dothome.co.kr/fileupload/up/data/rogo1.png";
                divItem[2] = "http://eor0601.dothome.co.kr/fileupload/up/data/rogo1.png";
                mineAdapter.addItem(new MineImage_Item(divItem[0], divItem[1], divItem[2]));
            }
        }else if(i%3 == 1) {
            divItem[1] = arrayStr[i];
            if(i==arrayLength-1){
                divItem[2] = "http://eor0601.dothome.co.kr/fileupload/up/data/rogo1.png";
                mineAdapter.addItem(new MineImage_Item(divItem[0], divItem[1], divItem[2]));
            }
        }else if(i%3 == 2) {
            divItem[2] = arrayStr[i];
            mineAdapter.addItem(new MineImage_Item(divItem[0], divItem[1], divItem[2]));
        }
    }
    mineUserList.setAdapter(mineAdapter);
}
```

App 설정

build.gradle

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.ezen.sapoom"
        minSdkVersion 17
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    //noinspection GradleCompatible
    implementation 'com.android.support:appcompat-v7:28.0.0-alpha1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.2'
    implementation 'com.android.support:support-v4:28.0.0-alpha1'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support:design:28.0.0-alpha1'

    // 볼리
    compile 'com.android.volley:volley:1.1.0+'

    // 피카소
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile files('libs/picasso-2.5.2.jar')
```

소스 파일 정리

프래그먼트 및 리스트 뷰	
로그인 DB 통신 기본 화면	

주로 사용한 PHP & MySQL 을 사용한 DB 서버 통신

- DB 테이블 구성은 5개를 생성

1. users [회원 가입 및 개인 정보 수정]

+ 옵션

←T→	user_index	db_nickname	db_id	db_pw	profile_sub	e_mail	profile_image
수정 복사 삭제	36	홍길이	hongkildong	123	정의로운 인물입니다.	eor0601@hanmir.com	a2c3f799bc0472abd36a97d366280fa8.png
수정 복사 삭제	52	나프카	haneor	123	나프카 잘부탁 드립니다.	eor0601@hanmir.com	55e8d8a4db956a2a5b8e12d34ac8f9af.png
수정 복사 삭제	53	홍길동	hong	123	저는 길동이 입니다	eor0601@hanmir.com	8cfb041cdd1ae6d98956b244ae1e7676.png
수정 복사 삭제	54	알파	abcd	123	abcdefg	eor0601@naver.com	db5f17843d579b775adbddee70690297.png
수정 복사 삭제	55	윤하늘	yhn	123	윤하늘 작가 입니다	eor0601@hanmir.com	0c1ee81daa5242ac25fe355acfca512d.png
수정 복사 삭제	56	나프	nafca	123	NULL	NULL	NULL

모두 체크 선택한 것을: 수정 복사 삭제 내보내기

2. update file [원본 이미지 저장 시 생성 및 삭제]

+ 옵션

←T→	db_id	file_id	name_orig	name_save	reg_time
수정 복사 삭제	haneor	01e5fdd4b48e57bfe81e47dc3706ad48	1534135270666.png	ba0a5ebee09599fdcabd2450a80c4e34.png	2018-08-13 13:41:09
수정 복사 삭제	haneor	149ef7c4474ecdcd4d42ca6b10595d9e	1533908740373.png	831065fff2cc5ddbfb3ef03f524f162e.png	2018-08-10 22:45:26
수정 복사 삭제	haneor	2ced3537a1460b277a6de03cbefaf412	1534146803958.png	d8005d09ee121b69b6afe5dca2b43dd4.png	2018-08-13 16:53:20
수정 복사 삭제	haneor	35c0e1c2a950b2463e69589d8c32ae8d	1534135253922.png	d7119e88df3aecbd74d0e868ed2f8efc.png	2018-08-13 13:40:51
수정 복사 삭제	haneor	4861c44d157723337960b53e997490cd	1534138740694.png	eee9e45fa4cfb300a1dac9a9e9415b5d.png	2018-08-13 14:38:58
수정 복사 삭제	yhn	6bd0f251649babd3a24178dd09dd7e6c	1533903502734.png	8c11748ef018575bd2eb3cd826f0168d.png	2018-08-10 21:18:09
수정 복사 삭제	haneor	819cf0f423c0eedd080adaaa39c20e27	1533909251791.png	91b090ced8c334e60d2cac0d26eb7f81.png	2018-08-10 22:53:58
수정 복사 삭제	haneor	9130966f6b2c92f7203d366cde77d04e	1534136465880.png	65070ef6cbcadf3037c289d5b96b8561.png	2018-08-13 14:01:02
수정 복사 삭제	hong	9d1fbae7f90b9029ccc8b6aa77497bb4	1533888710187.png	2924daa1fb5e78856dc2f4979d0f8e35.png	2018-08-10 17:11:35
수정 복사 삭제	hongkildong	a0010aafba8261b580799c79868beffd	1533888149259.png	f6a76dcf49bae156f6df5fb902e6c349.png	2018-08-10 17:02:15

3. image [사진 업로드 시 생성, 내용 변경 시 수정]

+ 옵션

←T→	image_index	db_id	db_substance	created	image_id	comment_sub	good
수정 복사 삭제	50	hongkildong	인물화입니다	2018-08-10 13:32:23	f6a76dcf49bae156f6df5fb902e6c349.png		0
수정 복사 삭제	51	hong	여전사	2018-08-10 13:41:42	2924daa1fb5e78856dc2f4979d0f8e35.png		0
수정 복사 삭제	52	yhn	첫번째 작품입니다 .	2018-08-10 16:33:50	beae0aa6addd7619b1baf67cd52738c6.png		0
수정 복사 삭제	53	yhn	옷이 참 이쁘다.	2018-08-10 17:48:31	8c11748ef018575bd2eb3cd826f0168d.png	5	5
수정 복사 삭제	54	hongkildong	날개	2018-08-10 18:41:43	dca70b7e1b60c2530f3039b640939397.png		0
수정 복사 삭제	55	haneor	냐옹	2018-08-10 19:15:33	831065fff2cc5ddbfb3ef03f524f162e.png		1
수정 복사 삭제	56	abcd	미인	2018-08-10 19:17:00	f64eec93d62003a75ebe4a9fb85a595c.png	4	4
수정 복사 삭제	57	haneor	시간	2018-08-10 19:24:03	91b090ced8c334e60d2cac0d26eb7f81.png	2	2
수정 복사 삭제	58	haneor	ㅋㅋㅋㅋ	2018-08-13 10:10:56	d7119e88df3aecbd74d0e868ed2f8efc.png		0
수정 복사 삭제	59	haneor	ㅋㅋㅋㅋㅋ	2018-08-13 10:11:14	ba0a5ebee09599fdcabd2450a80c4e34.png		0

4. follow [친구 추가 및 삭제 관리]

+ 옵션

←T→	follow_index	my_id	follow_id	created
수정 복사 삭제	40	yhn	hong	2018-08-10 15:49:08
수정 복사 삭제	41	hong	yhn	2018-08-10 17:49:08
수정 복사 삭제	42	haneor	yhn	2018-08-10 18:07:57
수정 복사 삭제	43	haneor	hong	2018-08-10 18:07:59
수정 복사 삭제	44	hongkildong	yhn	2018-08-10 18:41:49
수정 복사 삭제	45	hongkildong	hong	2018-08-10 18:41:52
수정 복사 삭제	46	abcd	yhn	2018-08-10 19:17:12
수정 복사 삭제	47	abcd	hong	2018-08-10 19:17:15
수정 복사 삭제	48	haneor	abcd	2018-08-10 19:36:29
수정 복사 삭제	49	haneor	hongkildong	2018-08-10 19:36:32

5. comment [정해진 사진에 대한 댓글을 저장]

+ 옵션

←T→	index	image_id	comment_sub	user_id	good	created
수정 복사 삭제	44	http://eor0601.dothome.co.kr/fileupload/up/data/f6...	안녕하세요 정말 그림을 잘그리시네요	haneor	1	2018-08-10 22:32:07
수정 복사 삭제	46	http://eor0601.dothome.co.kr/fileupload/up/data/f6...	작가님 이쁜 그림이 너무 좋아요	yhn	1	2018-08-12 17:54:48
수정 복사 삭제	47	http://eor0601.dothome.co.kr/fileupload/up/data/f6...	감사합니다. 더 좋은 작품 업로드 하도록 노력하겠습니다	abcd	0	2018-08-12 17:56:32
수정 복사 삭제	48	http://eor0601.dothome.co.kr/fileupload/up/data/8c...	항상 잘보고 있습니다	abcd	1	2018-08-12 18:02:45
수정 복사 삭제	49	http://eor0601.dothome.co.kr/fileupload/up/data/8c...	잘그리시네요ㅋㅋㅋ	hong	1	2018-08-12 18:04:00
수정 복사 삭제	50	http://eor0601.dothome.co.kr/fileupload/up/data/8c...	다음 작품은 언제쯤 올리시나요?	hongkildong	1	2018-08-12 18:05:11
수정 복사 삭제	51	http://eor0601.dothome.co.kr/fileupload/up/data/8c...	색체가 너무 이뻐요	haneor	1	2018-08-12 18:06:37
수정 복사 삭제	52	http://eor0601.dothome.co.kr/fileupload/up/data/8c...	감사합니다 앞으로 더욱 열심히 하겠습니다	yhn	1	2018-08-12 18:08:15

DB 와 통신을 위한 php

- php 작성 후 Postman으로 테스트를 실시

insert

```php
<?php
header('Content-Type: application/json; charset=utf8');
$conn = mysqli_connect("-", "-", "-", "-");

  if (!$conn) {
      die("Connection failed: " . mysqli_connect_error());
  }
  if (!$conn->set_charset("utf8")) {
      exit();
  } else {}
  $db_id = $_POST["db_id"];
  $db_pw = $_POST["db_pw"];
  $db_nickname = $_POST["db_nickname"];
  $sql = "INSERT INTO users (db_id, db_pw, db_nickname)
VALUES ('$db_id', '$db_pw', '$db_nickname');";
  if (mysqli_multi_query($conn, $sql)) {
      echo "New records created successfully";
  } else {
      echo "Error: " . $sql . "<br>" . mysqli_error($conn);
  }
    $response = array();
    $response["success"] ="true";
  echo json_encode($response);
  mysqli_close($conn);
?>
```

update

```php
<?php
  $con = mysqli_connect("-", "-", "-", "-");

  //한글 깨짐 바꾸기
  if (!$con->set_charset("utf8")) {
    exit();
} else {
}

  $db_id = $_POST["id"];
  $db_nickname = $_POST["nickname"];
  $profile_sub = $_POST["profile_sub"];
  $e_mail = $_POST["e_mail"];
  $profile_image = $_POST["profile_image"];
  mysqli_query($con, "UPDATE `users` SET db_nickname='$db_nickname',
   profile_sub='$profile_sub', e_mail='$e_mail',
    profile_image='$profile_image' WHERE db_id='$db_id'" );

     $response = array();
     $response["success"] = true;

  echo json_encode($response);

?>
```

select

```php
<?php
  header('Content-Type: application/json; charset=utf8'); // 업데이트 방식


  $con = mysqli_connect("-", "-", "-", "-");

  //한글 깨짐 바꾸기
  if (!$con->set_charset("utf8")) {
      exit();
  } else {
  }

  $db_id = $_POST["db_id"];
  $sql = "SELECT * FROM users WHERE db_id='$db_id'";
  $result = mysqli_query($con, $sql );
  $response = array();

  while($row = mysqli_fetch_array($result)){
    array_push($response, array("db_nickname"=>$row[1], "profile_sub"=>$row[4],
     "e_mail"=>$row[5], "profile_image"=>$row[6]));
  }
   echo json_encode(array("response"=>$response));
   mysqli_close($con);
?>
```

delete

```php
<?php
  header('Content-Type: application/json; charset=utf8');
  $con = mysqli_connect("-", "-", "-", "-");

  if($con === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
  }
    $image_path = $_POST["del_image_path"];
  $sql = "DELETE FROM upload_file WHERE name_save='$image_path'";
  $sqli = "DELETE FROM image WHERE image_id='$image_path'";

  if (mysqli_multi_query($con, $sql)) {
      echo "New records created successfully";
  } else {
      echo "Error: " . $sql . "<br>" . mysqli_error($con);
  }
  if (mysqli_multi_query($con, $sqli)) {
    echo "New records created successfully";
  } else {
    echo "Error: " . $sqli . "<br>" . mysqli_error($con);
  }
    $response = array();
    $response["success"] ="true";
  echo json_encode($response);
  mysqli_close($con);
?>
```

통신을 위한 Volley & Request 라이브러리 사용

- Volley & RequestQueue(String타입) = 주로 사용한 기능

```java
private void onLoadUpDateItem() {
    RequestQueue requestQueue = Volley.newRequestQueue( context: this);
    StringRequest stringRequest = new StringRequest(Request.Method.POST,  url: "http://eor0601.dothome.co.kr/loadupdate.php", new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonObject_response = new JSONObject(response);
                JSONArray jsonArray = (JSONArray) jsonObject_response.get("response");
                for (int i = 0; i < jsonArray.length(); i++) {
                    JSONObject result = (JSONObject) jsonArray.get(i);
                    horize_id_list.add(result.getString( name: "db_id"));
                    horize_image_list.add(result.getString( name: "profile_image"));
                    // horizelist 에 있는 정보를 가져와서 내부에 저장
                    HLVAdapter.profile_image_horizelist.add(IMAGE_URL + horize_image_list.get(i));
                    HLVAdapter.addfollow_id_list.add(horize_id_list.get(i));
                    for (int j = 0; j < follow_count; j++) {
                        if (follow_id[j].equals(result.getString( name: "db_id")) || follow_id[j].equals(LoginActivity.db_id)) {
                            image_db_id_list.add(result.getString( name: "db_id"));
                            userImage_list.add(result.getString( name: "image_id"));
                            substance_list.add(result.getString( name: "db_substance"));
                            comment_list.add(result.getString( name: "comment_sub"));
                            good_list.add(result.getString( name: "good"));
                            nickname_list.add(result.getString( name: "db_nickname"));
                            userProfileImage_list.add(result.getString( name: "profile_image"));
                            getFollow_count++;
                        }
                    }
                }
                updateListView();
                horizeListAdd();
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

        }
    }) {
        @Override
        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<~>();
            return params;
        }
    };
    requestQueue.add(stringRequest);
```

- Volley MultiRequest

```java
private void uploadBitmap(final Bitmap bitmap) {
    VolleyMultipartRequest volleyMultipartRequest = new VolleyMultipartRequest(Request.Method.POST, BASE_URL,
            new Response.Listener<NetworkResponse>() {
                @Override
                public void onResponse(NetworkResponse response) {
                    try {
                        // 프로필 수정 시 업로드된 사진을 피드백 받아서 설정함.
                        String responsePath = new String(response.data, charsetName: "UTF-8");
                        // 저장 시 버그로 ""의 포함으로 버그 발생하여 처리함.
                        image_id = responsePath.substring(1, 37);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    // 만약 업데이트 사진의 주소를 확보 하려면 IMAGE_URL + image_id (String) 으로 받아 오면 됨.
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Toast.makeText(getApplicationContext(), error.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }) {

        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<>();
            db_id = LoginActivity.db_id;
            params.put( k: "db_id", db_id);
            return params;
        }

        protected Map<String, DataPart> getByteData() {
            Map<String, DataPart> params = new HashMap<>();
            long imagename = System.currentTimeMillis();
            params.put( k: "upfile", new DataPart( name: imagename + ".png", getFileDataFromDrawable(bitmap)));
            return params;
        }
    };
    Volley.newRequestQueue( context: this).add(volleyMultipartRequest);
}
```

주요 사용 코드 정리

1. RecyclerView (horizontal) 사용
2. 이미지 및 각종 UI 구성은 listView로 사용
3. 이미지 라운드 처리 (CircleimageView) 사용
4. 간편한 이미지 처리를 위한 Picasso 사용
5. 서버에 올려진 이미지를 라운드 처리 하는 메소드 사용
6. 이미지 처리 시 메모리 사용을 줄이기 위해 이미지 용량을 줄여주는 메소드 사용

7. DB 통신을 위한 Volley 라이브러리 사용
8. 작성한 php를 확인하기 위한 POSTMAN 사용
9. 여러 DB 테이블과 php를 이용한 유저 정보 관리 로직 사용

10. onClick/onLongClick 을 이용한 정보 삭제 및 화면 전환 기능 사용
11. searchView를 이용한 검색 기능 사용
12. 애드몹(AdMob) 광고 연동

어플리케이션 정보

개발 환경 : 안드로이드 스튜디오
App 용량 : 4.5MB
App 마켓 주소 :
https://play.google.com/store/apps/details?id=com.han.sapoom&rdid=com.han.sapoom
필요한 Android 버전 : 4.2 이상